RDBMS for Environmental Observations

Bill Howe, Phd

UW eScience Institute

UW CSE

eScience Institute

http://escience.washington.edu

Web Images Videos Maps News Shopping Gmail more ▼

Google escience institute Search Advanced Search

Web Show options... Results **1** - **10** of about **908,000** for **escience** institute. (**0.40** seconds)

Did you mean: ***e science*** institute

University of Washington **eScience Institute**
The University of Washington **eScience Institute** was formed in January 2008 by the Vice President for UW Technology and the Provost, in partnership with key **...**
escience.washington.edu/ - Cached - Similar

University of Washington **eScience Institute**
For more information about the **Institute**, or for help with your data-intensive research, please contact: info **escience**.washington.edu **...**
escience.washington.edu/content/ - Cached - Similar

Show more results from escience.washington.edu

National **e-Science** Centre
e-Science Hub Kelvin Building Glasgow G12 8QQ United Kingdom Tel: 0141 330 8606. Fax: 0141 330 4913 email, **e-Science Institute** 15 South College Street **...**
Contacts - e-Science Events - About NeSC - Technical Papers
www.nesc.ac.uk/ - Cached - Similar

Web | Images | Videos | Shopping | News | Maps | More | MSN | Hotmail

bing

escience institute

ALL RESULTS

Were you looking for: ***science** institute*

ALL RESULTS

1-10 of 87,700,000 results · Advanced

RELATED SEARCHES

- Science **Museum**
- **Franklin** Institute Science **Museum**
- **Search** Institute
- Science **Center**
- **Skin** Science Institute
- **Reproductive** Science Institute
- **Health** Sciences Institute
- **Brick Computer** Science Institute

SEARCH HISTORY

- escience institute
- escience institute washington

Institute of Noetic **Sciences**: Home Page

IONS is a nonprofit membership organization that conducts and sponsors leading-edge research into the potentials and powers of consciousness: including perceptions, beliefs ...

www.noetic.org · Cached page

Virginia **Institute** of Marine **Science** - Home

An **institution** that is uniquely prepared to educate the highly qualified researchers, resource managers, and educators needed for the future. Gloucester Point, Virginia.

www.vims.edu · Cached page

ICSI: International Computer **Science Institute** | Berkeley, California

International Computer **Science Institute** at the University of California, Berkeley. Research groups and technical reports.

http.icsi.berkeley.edu · Cached page

National **Institute** for Discovery **Science** (NIDS): **Science institute** ...

The National **Institute** for Discovery **Science** (NIDS) is a privately funded **science institute** engaged in research of UFOs, animal mutilations, and other related anomalous phenomena.

www.nidsci.org · Cached page

Background

- Phd Computer Science
  - Databases
  - Scientific Data Management
  - Unstructured Grids
- Research Scientist at CMOP
  - One of 17 NSF STCs
  - Observations and Models
- Senior Research Scientist, UW eScience
- Affiliate Assistant Professor, CSE

R/V Wecoma, April 2007

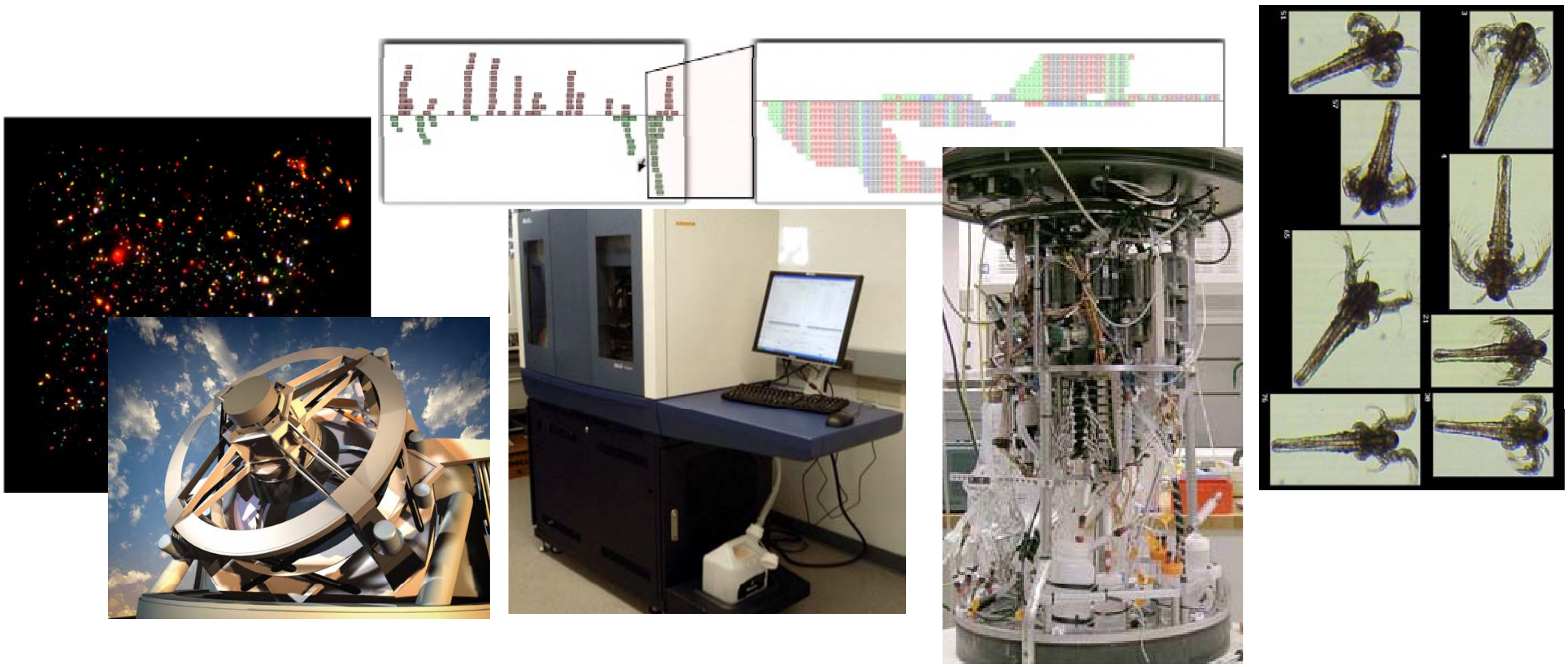
me

All science is reducing to querying databases

Old model: "Query the world" (Data acquisition coupled to a specific hypothesis)

New model: "Download the world" (Data acquired en masse, in support of many hypotheses)

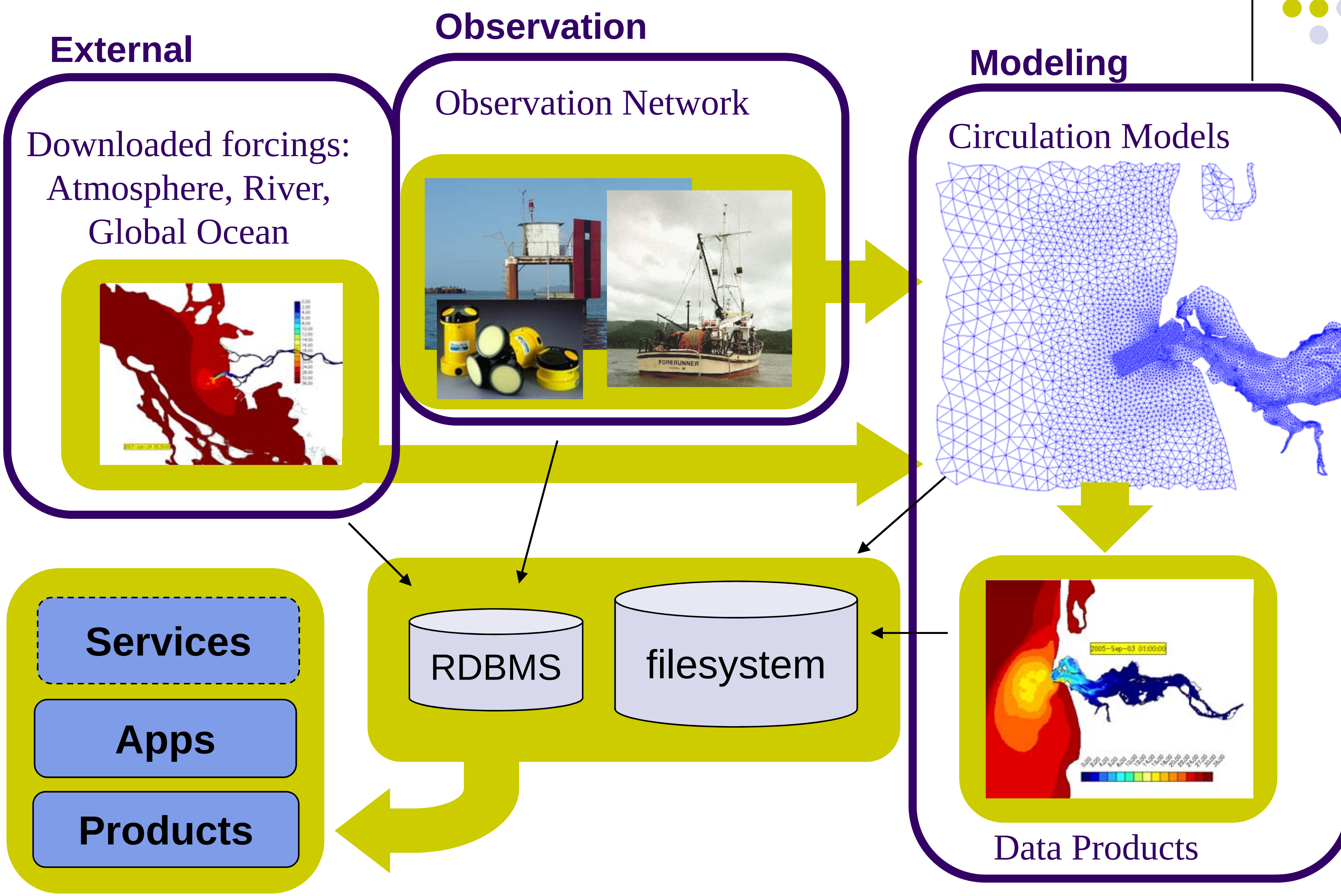
- Astronomy: High-resolution, high-frequency sky surveys (SDSS, LSST, PanSTARRS)
- Oceanography: high-resolution models, cheap sensors, satellites
- Biology: lab automation, high-throughput sequencing,



Topics

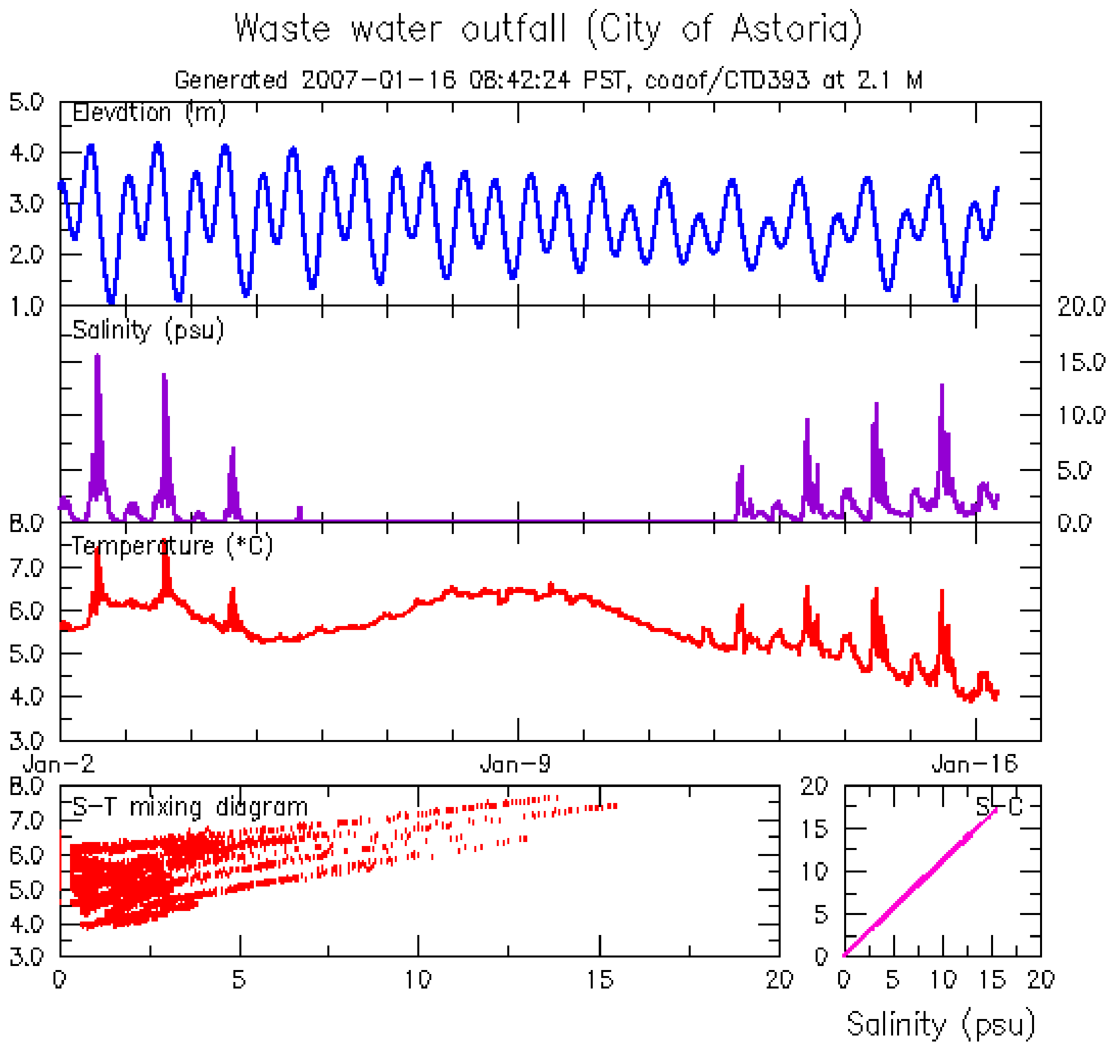
- Narrative: RDBMS at an Ocean Observatory
  - Strengths and Weaknesses
- Directions for discussion
  - Ad Hoc Databases
    - *Key discovery: SQL does not terrify scientists!*
  - Cloud computing & Database-as-a-service
  - Array-oriented Databases
  - Column-oriented Databases
  - Unstructured grids
  - Data-intensive Scalable Computing

Ex: CMOP Data Flow

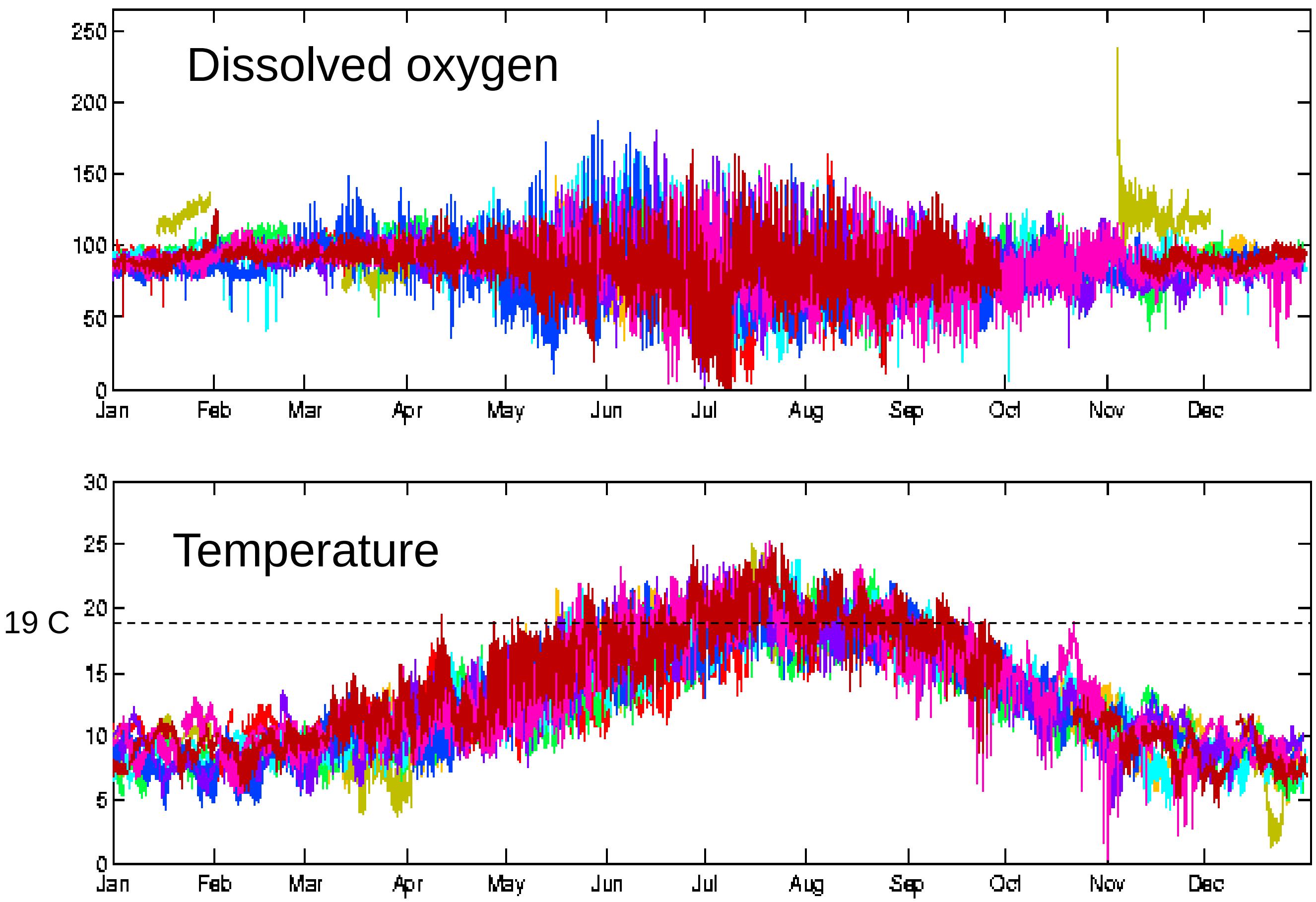


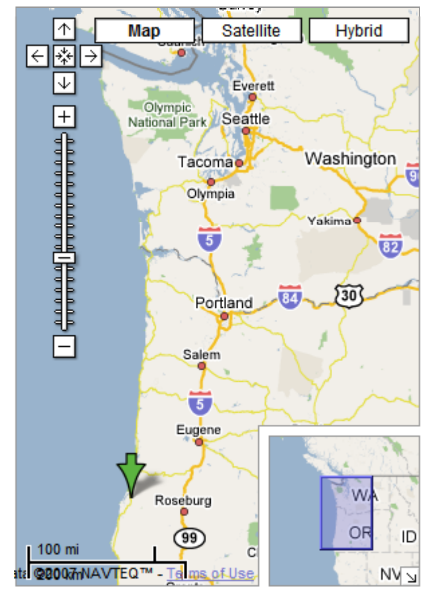
CMOP Observation Network

Example Products

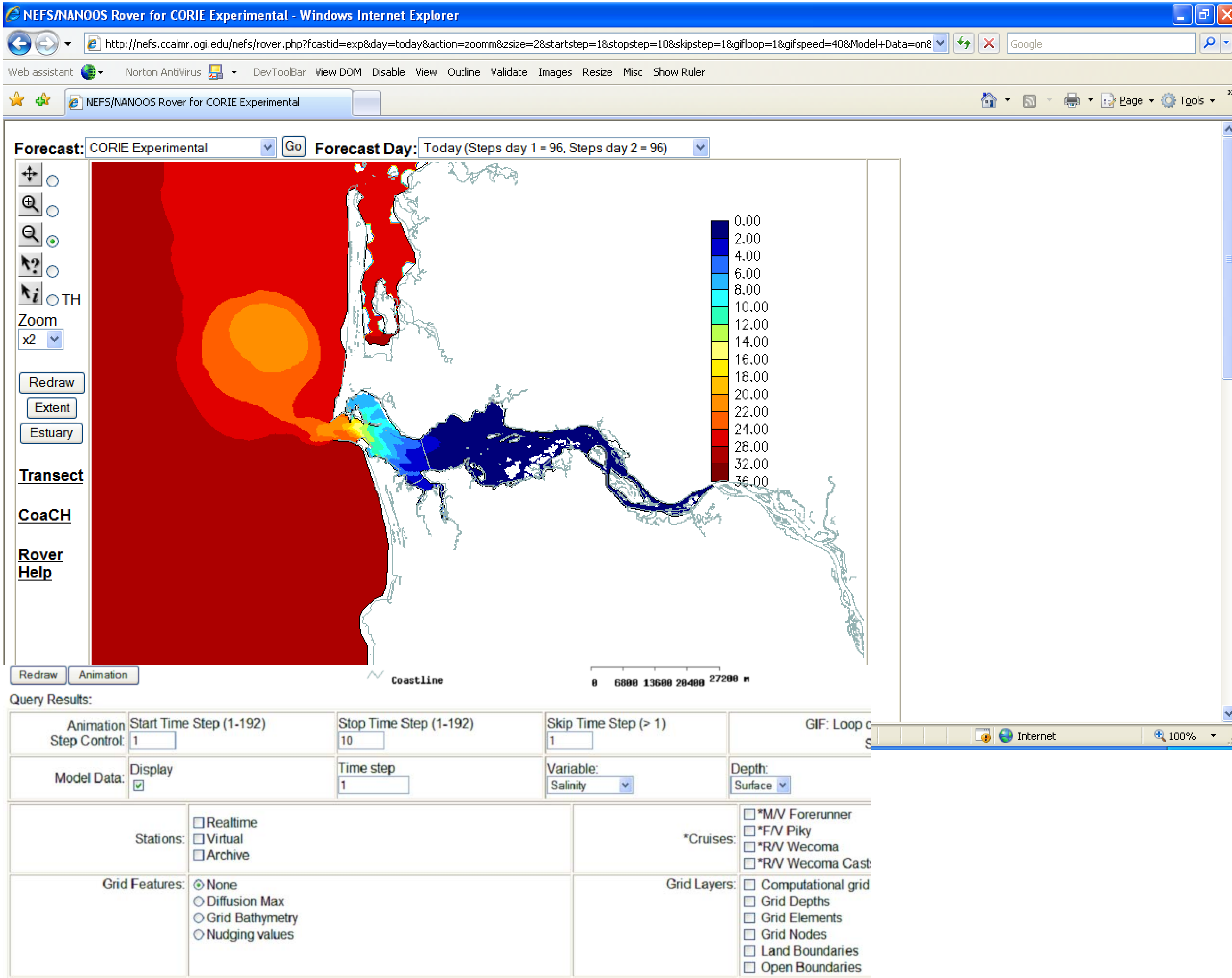


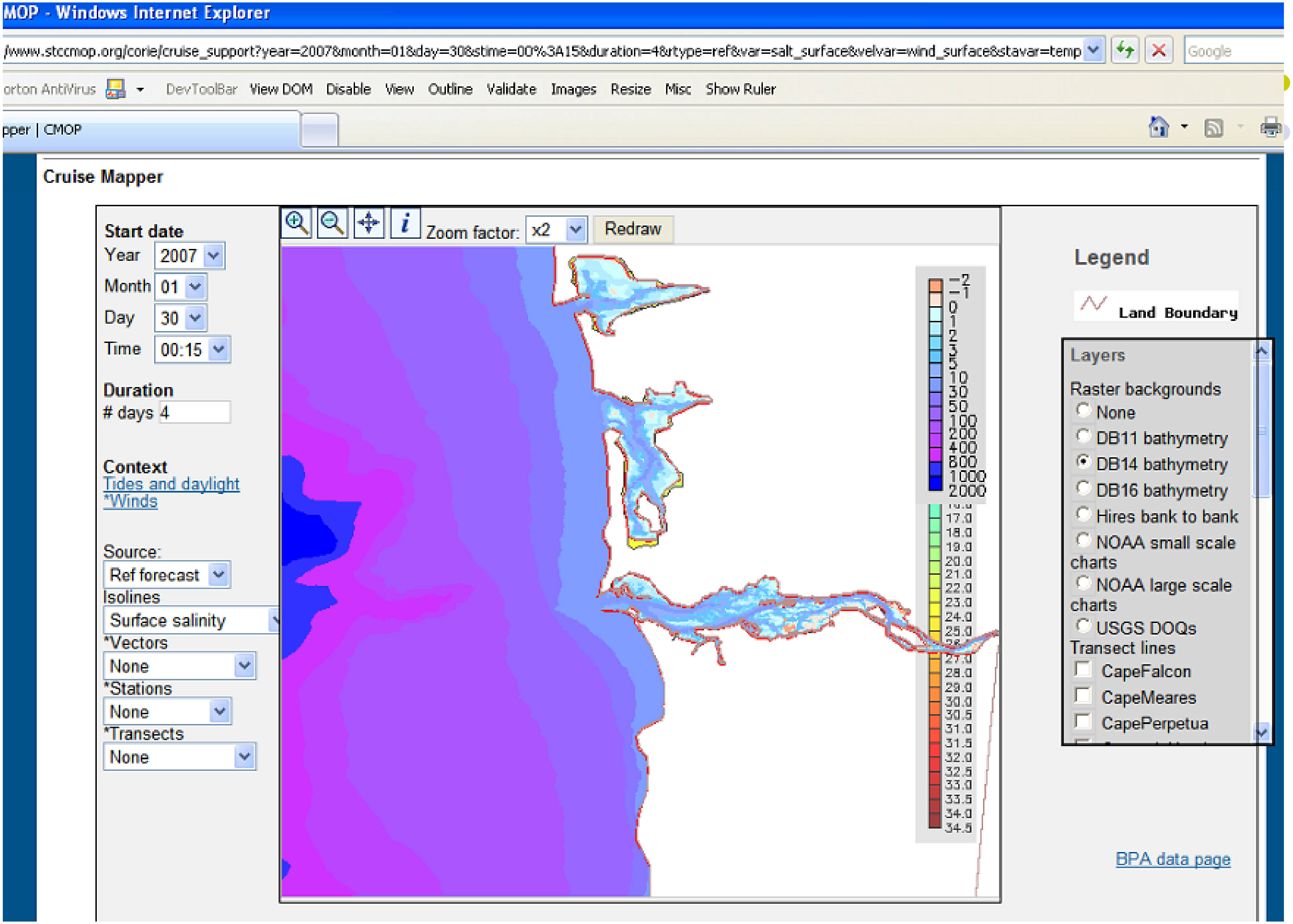
Example Products



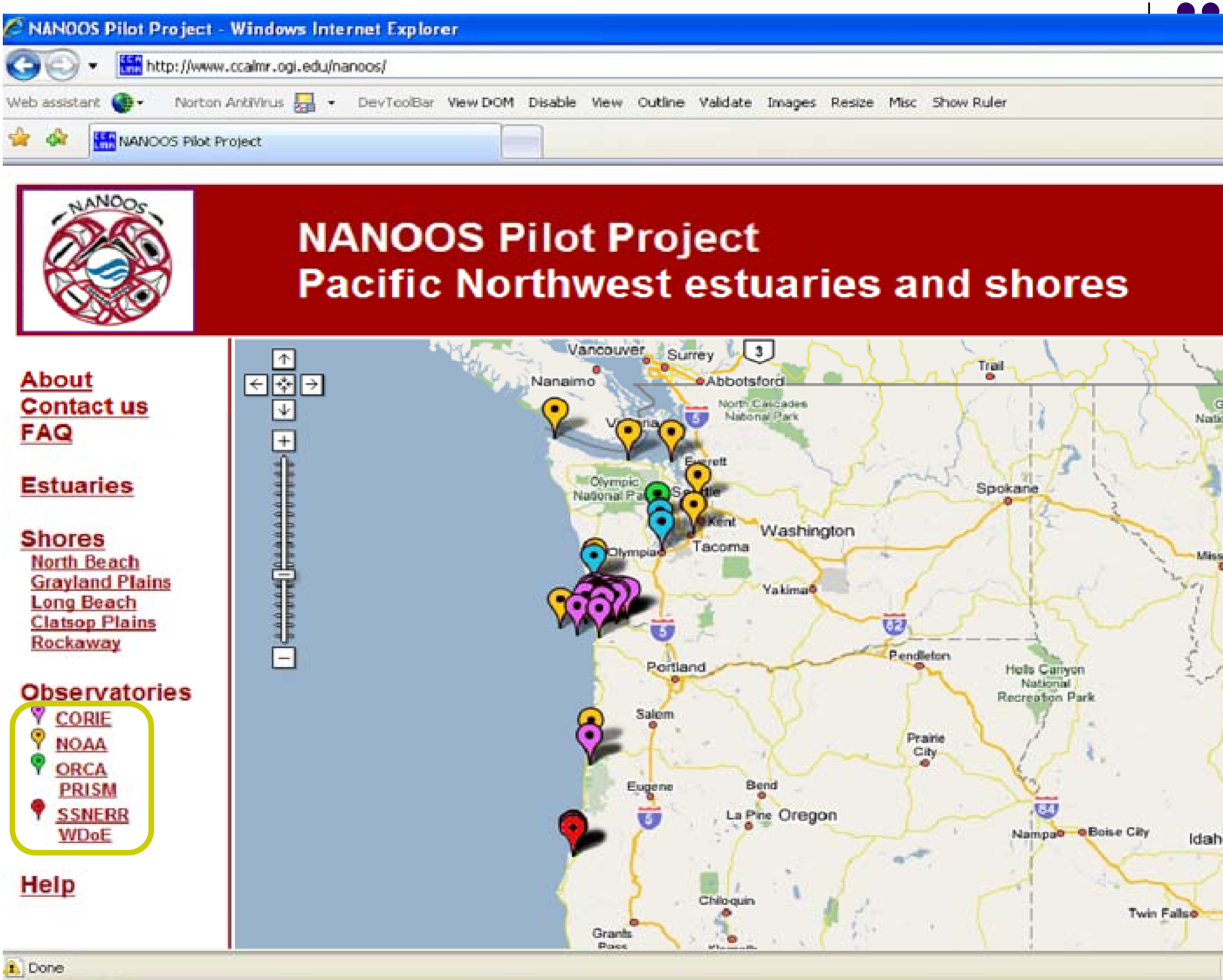


Yearly overlay scatter plot, 1996-2004, Winchester Arm, Coos Bay, OR


NEFS/NANOOS Rover for CORIE Experimental - Windows Internet Explorer
http://nefs.ccalmr.ogi.edu/nefs/rover.php?fcastid=exp&day=today&action=zoomm&zsize=2&startstep=1&stopstep=10&skipstep=1&gifloop=1&gifspeed=40&Model+Data=on&
Google
Web assistant
Norton AntiVirus
DevToolBar
View DOM
Disable
View
Outline
Validate
Images
Resize
Misc
Show Ruler
NEFS/NANOOS Rover for CORIE Experimental
Page
Tools
Forecast: CORIE Experimental
Go
Forecast Day: Today (Steps day 1 = 96, Steps day 2 = 96)
TH
Zoom
x2
Redraw
Extent
Estuary
Transect
CoaCH
Rover Help
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
20.00
22.00
24.00
28.00
32.00
36.00
Coastline
0 6800 13600 20400 27200 m
Redraw
Animation
Query Results:
Animation Step Control:
Start Time Step (1-192)
1
Stop Time Step (1-192)
10
Skip Time Step (> 1)
1
GIF: Loop c
Internet
100%
Model Data:
Display
Time step
1
Variable:
Salinity
Depth:
Surface
Stations:
Realtime
Virtual
Archive
*Cruises:
*M/V Forerunner
*F/V Piky
*R/V Wecoma
*R/V Wecoma Cast
Grid Features:
None
Diffusion Max
Grid Bathymetry
Nudging values
Grid Layers:
Computational grid
Grid Depths
Grid Elements
Grid Nodes
Land Boundaries
Open Boundaries

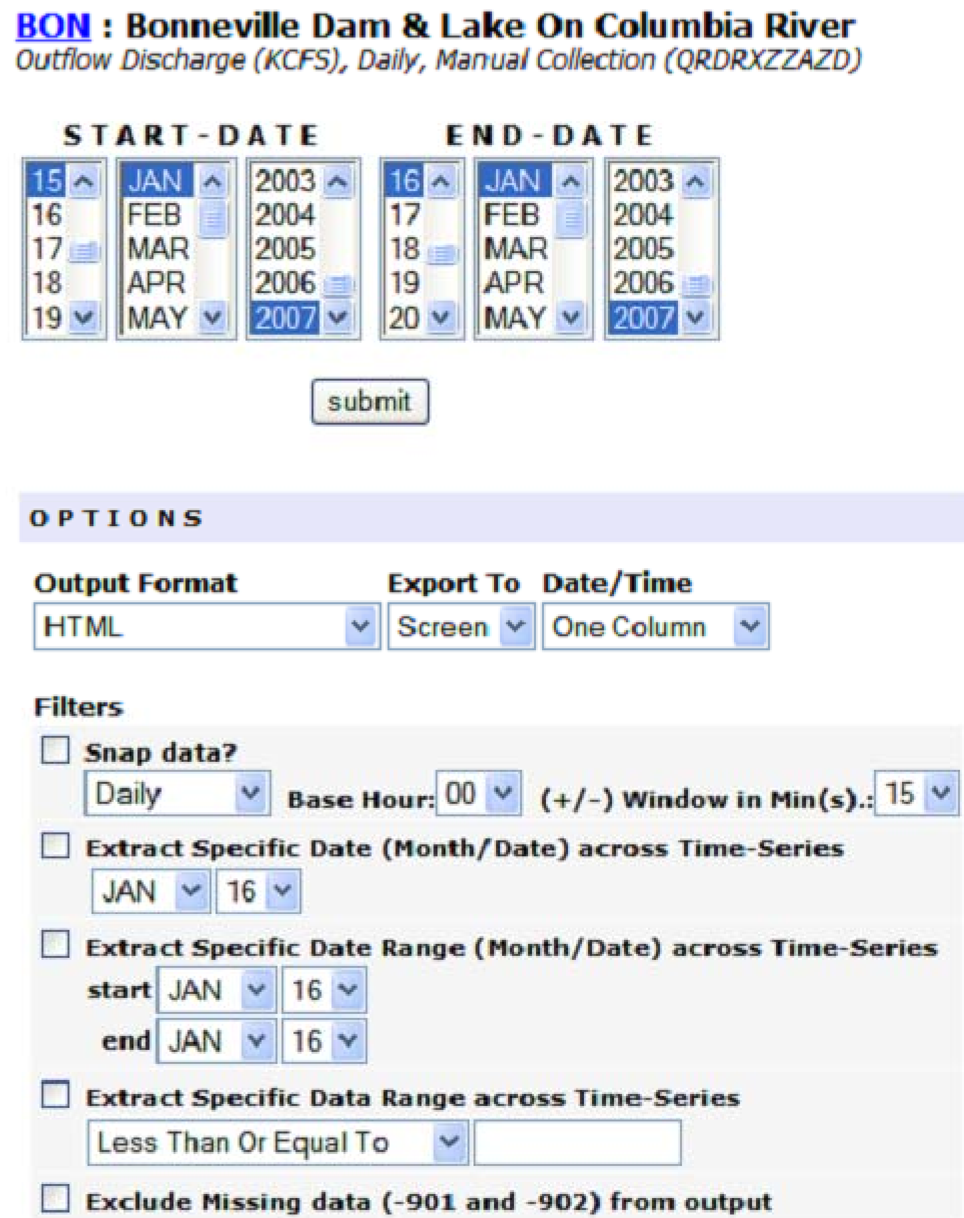


MOP - Windows Internet Explorer
/www.stccmop.org/corie/cruise_support?year=2007&month=01&day=30&stime=00%3A15&duration=4&rtype=ref&var=salt_surface&velvar=wind_surface&stavar=temp
orton AntiVirus
DevToolBar View DOM Disable View Outline Validate Images Resize Misc Show Ruler
pper | CMOP
Cruise Mapper
Start date
Year 2007
Month 01
Day 30
Time 00:15
Duration
days 4
Context
Tides and daylight
*Winds
Source:
Ref forecast
Isolines
Surface salinity
*Vectors
None
*Stations
None
*Transects
None
Zoom factor: x2
Redraw
Legend
Land Boundary
Layers
Raster backgrounds
None
DB11 bathymetry
DB14 bathymetry
DB16 bathymetry
Hires bank to bank
NOAA small scale charts
NOAA large scale charts
USGS DOQs
Transect lines
CapeFalcon
CapeMeares
CapePerpetua
BPA data page

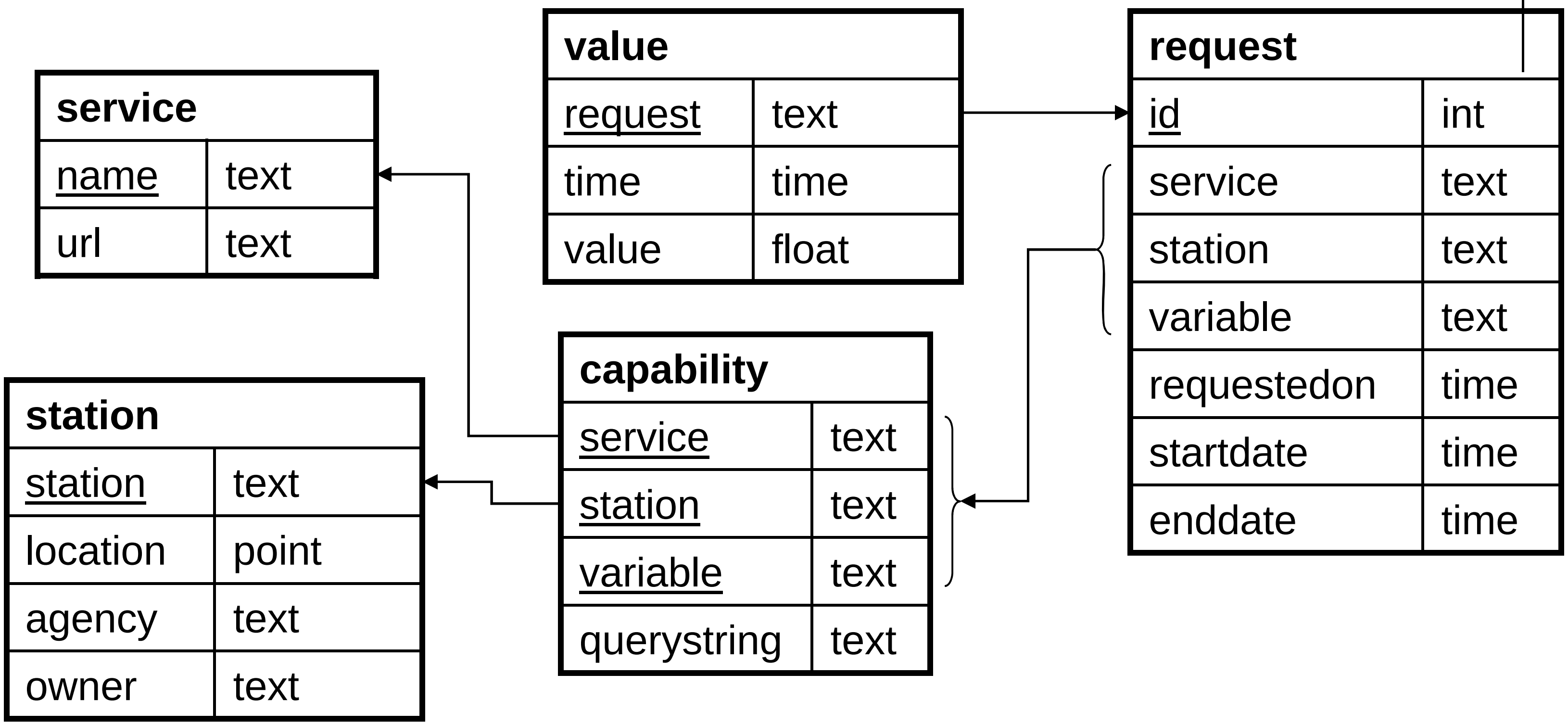

NANOOS Pilot Project - Windows Internet Explorer
http://www.ccalmr.ogi.edu/nanoos/
NANOOS Pilot Project
NANOOS
NANOOS Pilot Project
Pacific Northwest estuaries and shores
About
Contact us
FAQ
Estuaries
Shores
North Beach
Grayland Plains
Long Beach
Clatsop Plains
Rockaway
Observatories
CORIE
NOAA
ORCA
PRISM
SSNERR
WDoE
Help
Vancouver
Surrey
Nanaimo
Abbotsford
Trail
North Cascades National Park
Olympic National Park
Spokane
Washington
Tacoma
Olympia
Kent
Yakima
Portland
Pendleton
Hells Canyon National Recreation Park
Salem
Prairie City
Eugene
Bend
La Pine
Oregon
Nampa
Boise City
Chiloquin
Twin Falls
Done

External Data

- Model inputs
  - River discharge, River temperature, River elevation
  - Atmospheric Forcings
- Redistribute data from other institutions
- Fetched Daily
  - FTP
  - Screen scraping
  - CGI queries returning ASCII

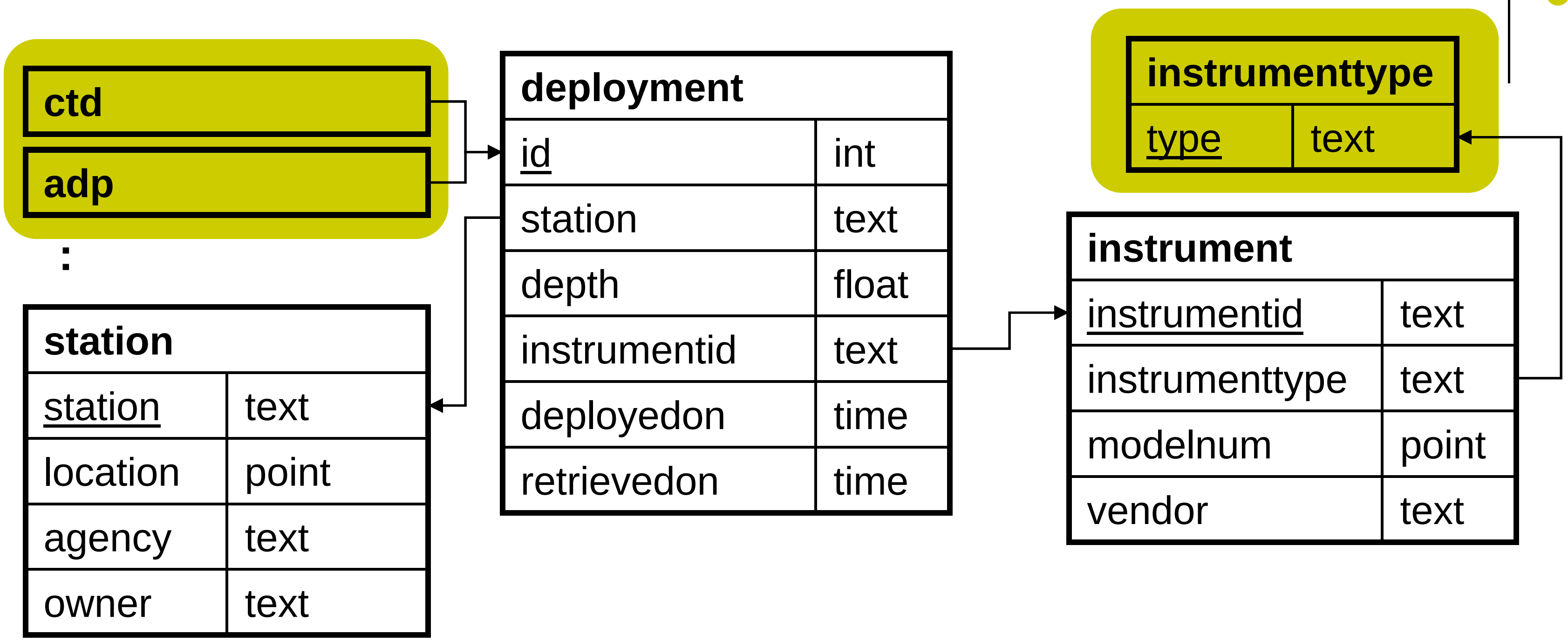


External Data



- Script assembles request url and parses result
  - one script per webservice
- Requests may overlap
- Redundant values are filtered out and occasionally deleted

Observations at Fixed Stations



- Tables for specific instrument types
- Deployment is a relationship between a fixed station, an instrument, and a range of time

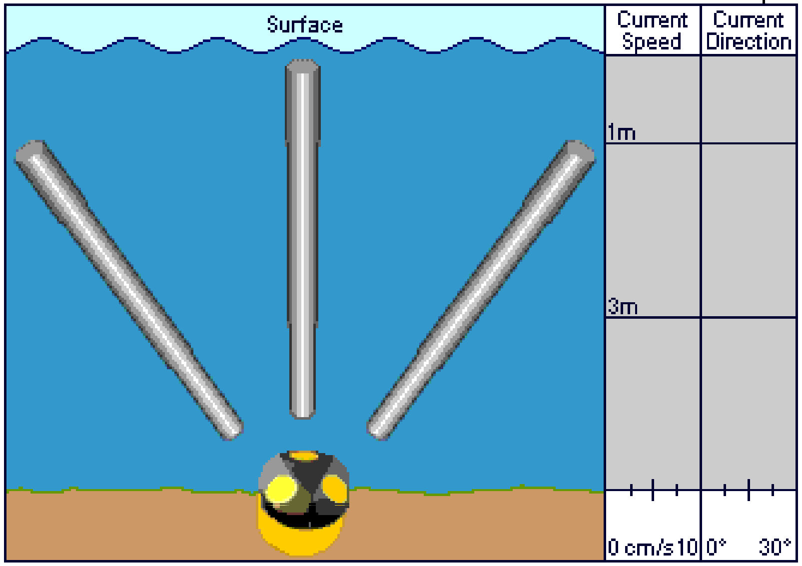
CTD:
Conductivity, Temperature, Depth

- deployment id (FK)
- time
- conductivity, temperature, pressure
- salinity, depth
- battery
- …
- rawrecord

ADP:
Acoustic Doppler Profiler

adp	
<u>deploymentid</u>	int
<u>time</u>	int
elevation	float
soundspeed	float
pitch	float
roll	float
…	

adpbin	
<u>adp</u>	text
depth	point
v[3]	text
vstd[3]	text



MET:
Meterological Station

met	
deploymentid	int
time	int
temperature	float
pressure	float
windspeed	float
winddirection	float
gust	float
humidity	float
rawrecord	text

Generalizing

- Data adequately captured, but many queries are awkward and inefficient
- Ex: “What’s the temperature near my ship?”
- Requires a union of model extractions, instrument readings, and downloaded datasets

Generalizing

request	
id	int
service	text
station	text
variable	text
requestedon	time
startdate	time
enddate	time

value	
request	text
time	time
value	float

deployment	
id	int
station	int
depth	float
instrumentid	text
deployedon	time
retrievedon	time

met	
deploymentid	int
time	int
temperature	float
pressure	float
windspeed	float
winddirection	float
gust	float
humidity	float
rawrecord	text

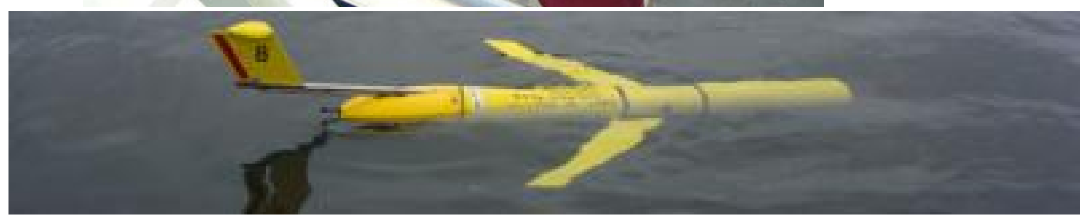
result	
resultid	text
runid	text
variable	text

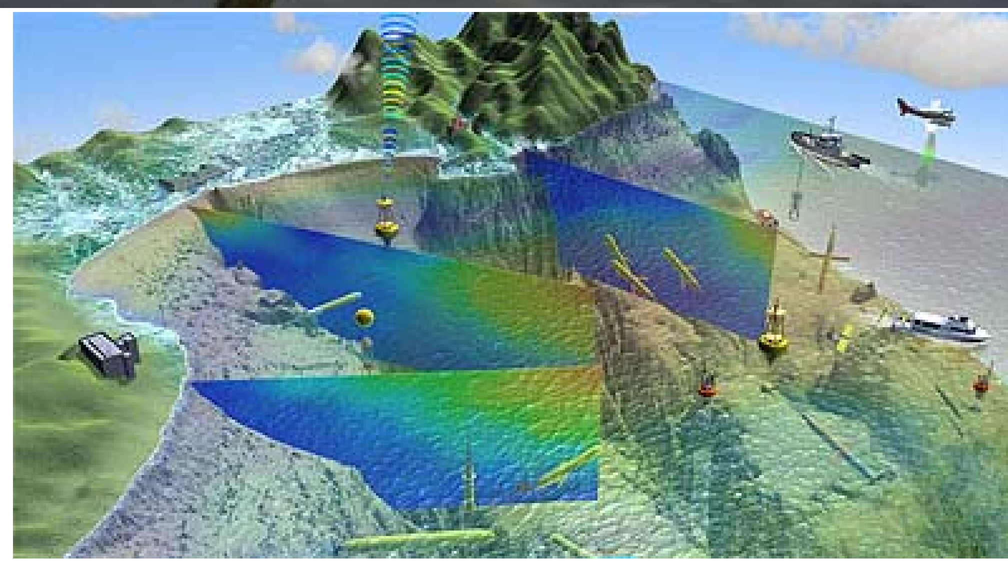
extraction	
resultid	text
value	float

Generalizing (2)

	dataset	
	id	int
(x,y)	station	int
	variable	text
(z)	depth	float
	starttime	time
	endtime	time

	value	
	datasetid	int
(t)	time	time
(v)	value	float



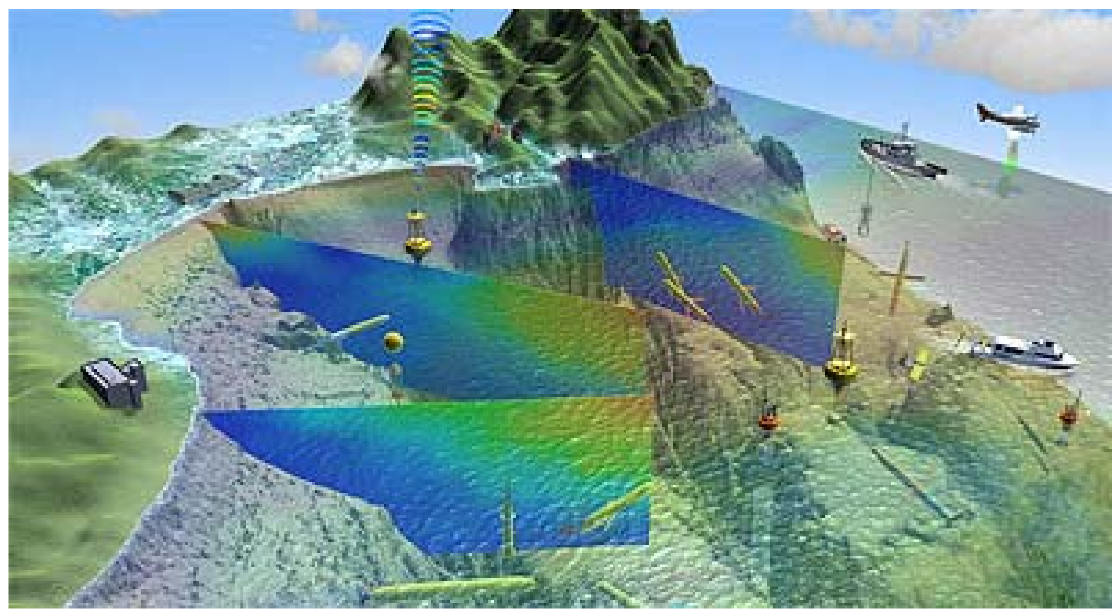


Generalizing (3)

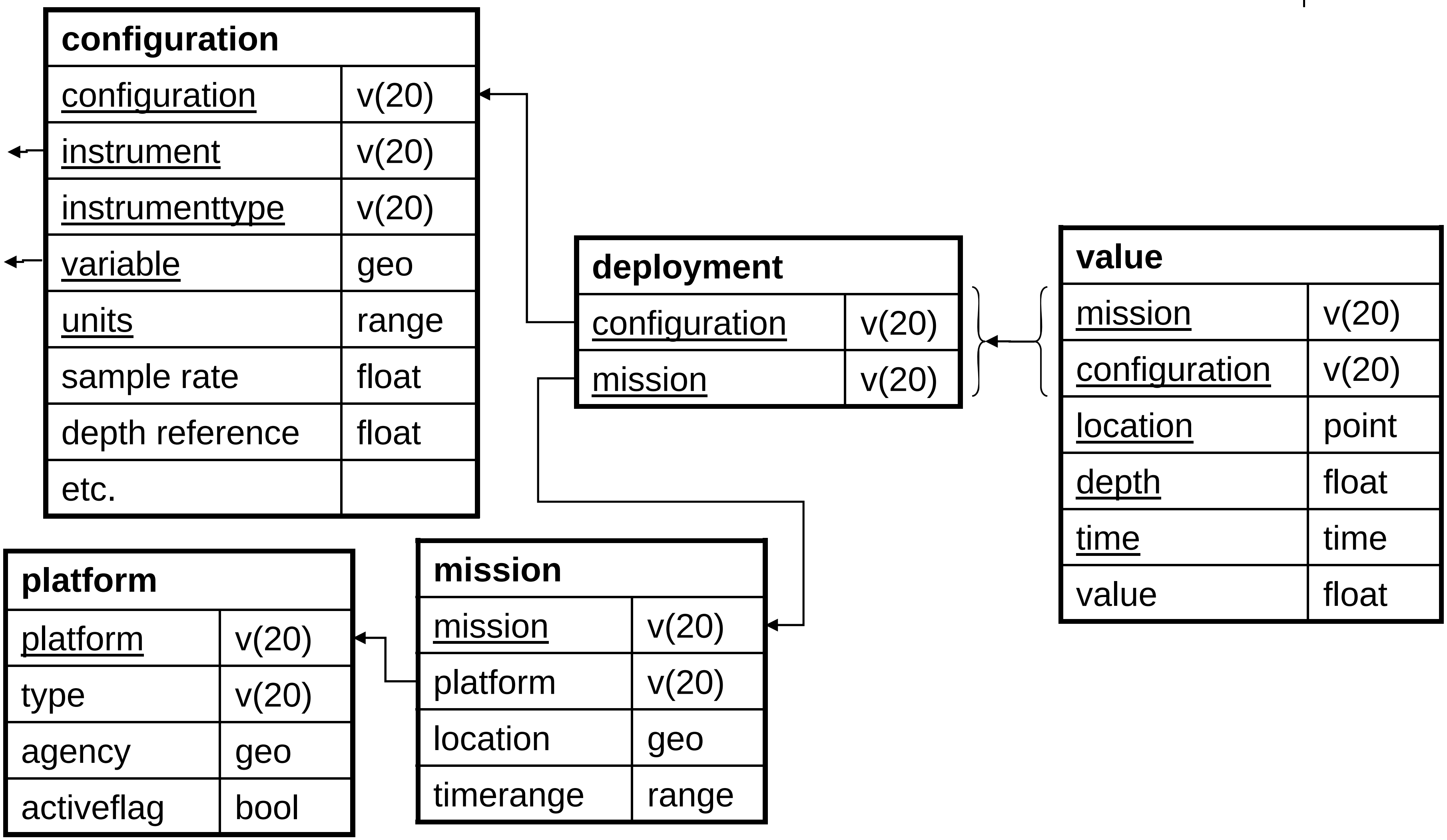
dataset	
datasetid	int
variable	text
units	text

tag	
datasetid	int
property	text
value	text

	value	
	datasetid	int
(x,y)	location	point
(z)	depth	float
(t)	time	time
(v)	value	float



Integrated Schema



Example: Fixed Stations

- **Platform**
  - One per station, e.g., ‘red26’
- **Mission**
  - One begins at each visit
  - Stores a spacetime ‘Envelope’; a single point in this case.
- **Configuration**
  - One per (instrument, variable)
  - Can be reused on multiple missions
- **Deployment**
  - assignment of a configuration to a mission
- **Value**
  - measurements redundantly store the location information

Example: Forerunner

- **Platform**
  - The forerunner itself,
  - maybe the cast devices are separate
- **Mission**
  - One for each cruise
  - Envelope is the cruise plan as a PostGIS linestring
- **Configuration / Deployment**
  - Sample rate, cast protocol, etc.
- **Value**
  - Time/Location changes with each measurement

Example: Oceanic Remote Chemical-optical Analyzer

- **Platform**
  - The ORCA mooring itself
- **Mission**
  - One for each of 3 long-term deployments
  - Envelope is a single point
- **Configuration / Deployment**
  - Sample rate, cast protocol, etc.
- **Value**
  - Location: local drift is ignored
  - Time: estimated from sample rate and profile time
  - Depth: estimated from pressure and latitude with a postgres UDF

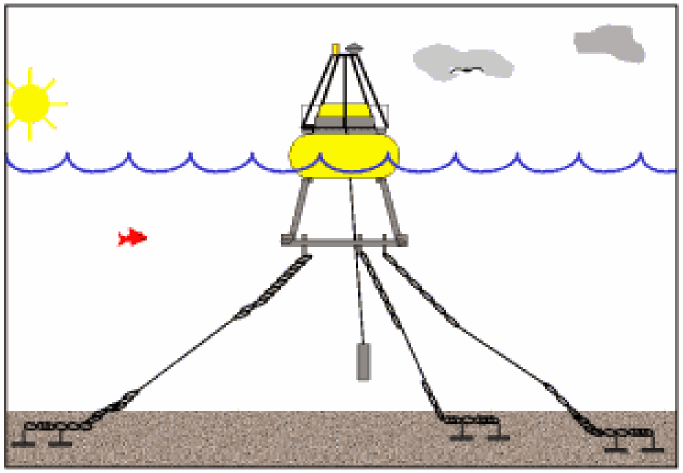


image source: http://www.ocean.washington.edu/research/orca/

Example: Umanned Underwater Vehicles

- **Platform**
  - The vehicle
- **Mission**
  - Each excursion is a mission
  - Envelope is a linestring
- **Configuration / Deployment**
  - Frequent changes
- **Value**
  - Location, depth, and time all vary

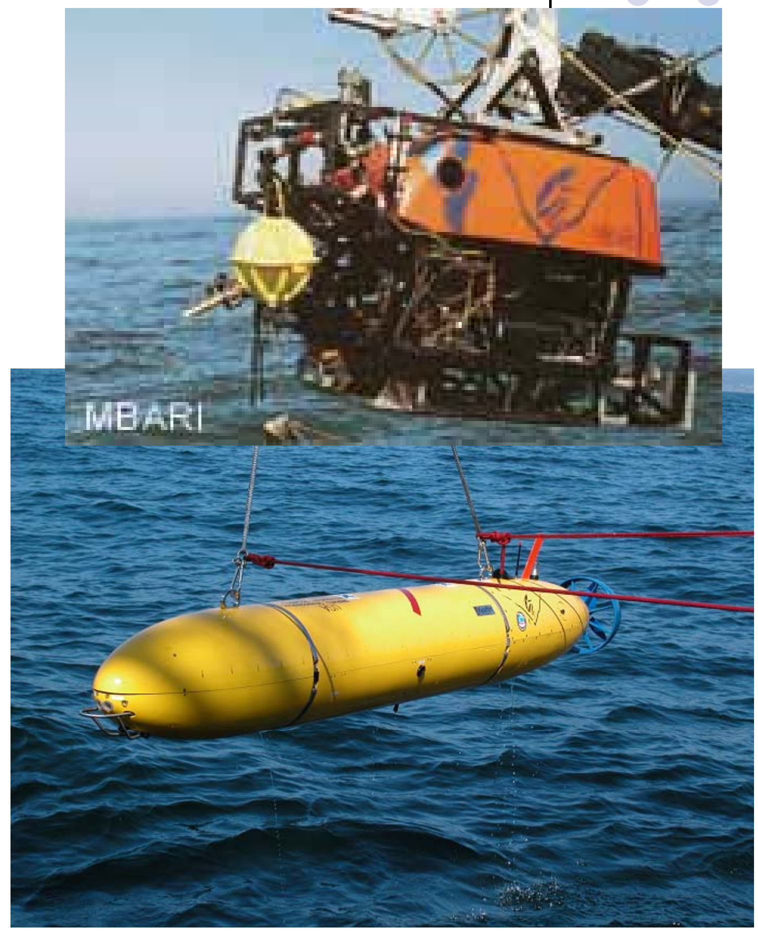
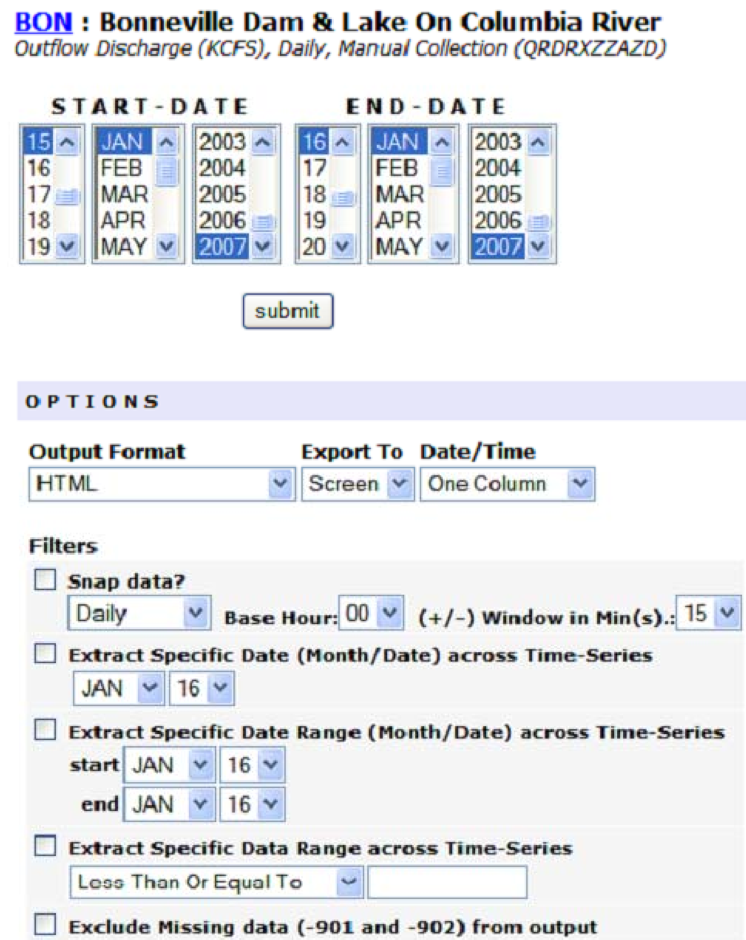


image source: http://www.mbari.org/auv/

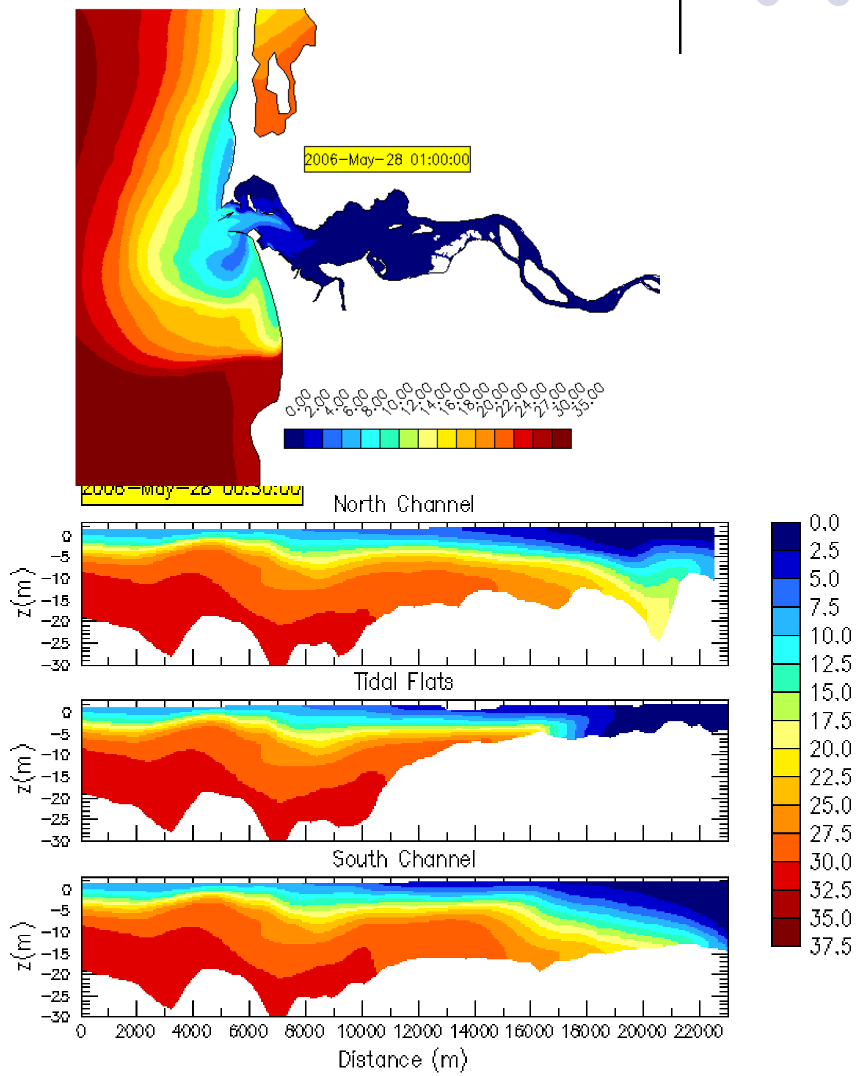
Example: External Data

- **Platform**
  - Each site or service
- **Mission**
  - Fine-grained: each request
  - Coarse-grained: each feed
- **Instrument**
  - Program used for retrieval
- **Configuration / Deployment**
  - Command-line arguments
- **Value**
  - Location, depth, and time may vary



Example: Model Extractions

- **Platform**
  - Each model: ELCIRC, SELFE
- **Mission**
  - A sequence of related runs
  - e.g, “Tilamook forecast”
- **Instrument**
  - “Virtual Instruments”
- **Configuration / Deployment**
  - Command-line arguments
- **Value**
  - Location, depth, and time may vary



Model Extractions Only

~1M values per timestep

* 96 timesteps per day

* 365 days per year

* ~8 years

* ~100 bytes

= 28 TB per “mission”

Containment in SpaceTime

- Platform may have a fixed location

  e.g., permanent moorings

  - If not, the mission may be to a fixed location

    e.g., ORCA

    - If not, the deployment may be at a fixed location

      e.g., Casts

      - If not, each measurement may be at a different location

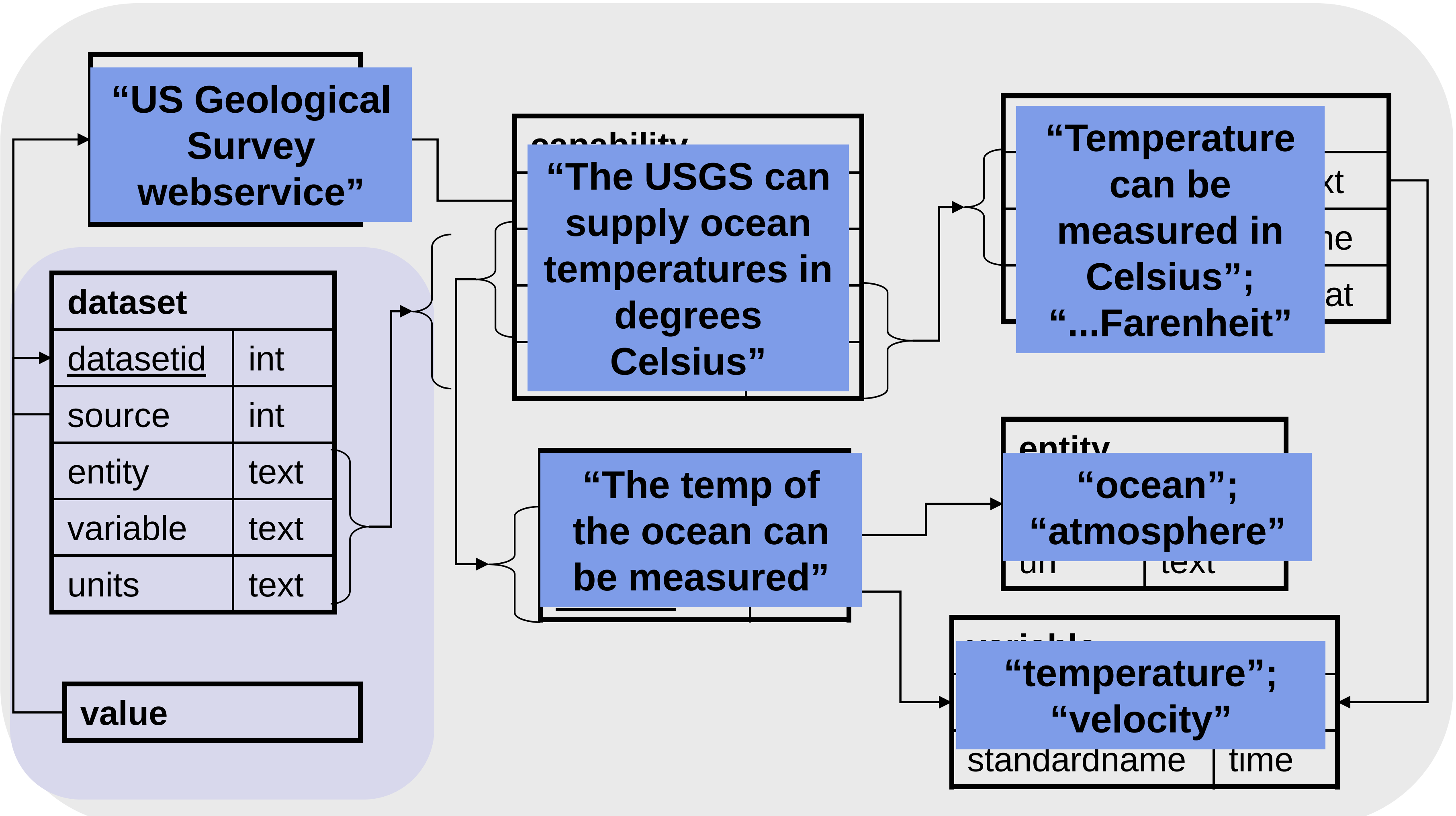
        e.g., UUVs

Defaults cascade upwards

Parent location contains its children's locations if not null.

Question: Implementation in PostgreSQL?

Other Generalizations: Measurable Quantities



Other Generalizations: Unit Conversions

units	
variable	text
units	time
standardname	float

$T = a*F + b$

linearconversion	
fromvariable	text
fromunit	text
tovariable	text
tounit	text
a	float
b	float

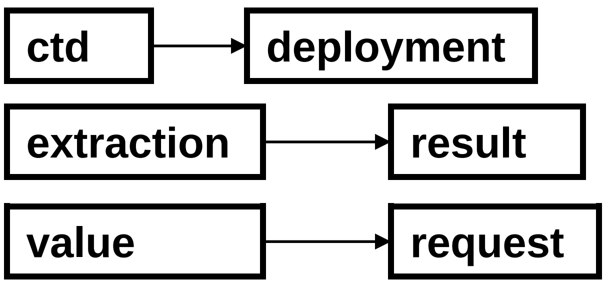
dataset	
datasetid	int
variable	text
units	text

value	
datasetid	int
location	point
depth	float
time	time
value	float

```
SELECT v.time,
       to.a*v.value + to.b as value
  FROM dataset d,
       linearconversion c,
       value v
 WHERE d.datasetid = v.datasetid
   AND d.variable = c.fromvariable
   AND d.units = c.units
   AND c.tovariable = ‘temp’
   AND c.tounit = ‘C’
```

Tradeoffs

schema complexity




```
SELECT c.salt
  FROM deployment d, ctd c
 WHERE c.deployment = d.id
   AND c.time BETWEEN ? AND ?
   AND d.station = ‘red26’
```

application complexity


```
SELECT v.value
  FROM dataset d,
       tag t,
       value v
 WHERE t.dataset = d.id
   AND v.dataset = d.id
   AND t.name = ‘station’
   AND t.value = ‘red26’
   AND d.variable = ‘salt’
   AND v.time BETWEEN ? AND ?
```

Recap

- Schema design is the fundamental activity
  - An expensive exercise
  - little intrinsic value
  - Poor adaptability
  - Permanently too general or too specific
  - Same goes for ontology/controlled vocabulary/standard
- Some queries are slow and complicated
  - Sets are good for ad hoc selection
  - Arrays are better for “neighborhood” queries
    - 1d Example: When did the temperature drop by more than 3 degrees from the previous measurement?
- But several useful capabilities (next slides)

Relational Database History

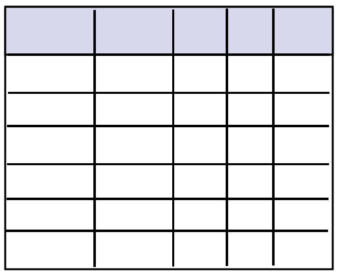
Pre-Relational: if your data changed, your application broke.

Early RDBMS were buggy and slow (and often reviled), but required only 5% of the application code.

"Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed." *-- Codd 1979*

Key Ideas: Programs that manipulate tabular data exhibit an algebraic structure allowing reasoning and manipulation independently of physical data representation

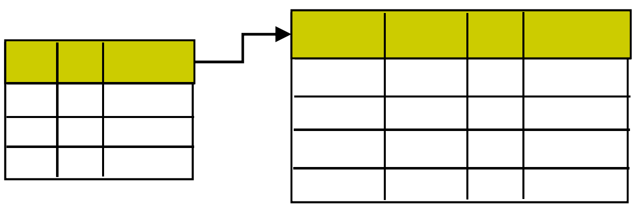
Key Idea: Data Independence



views

```
SELECT *
  FROM my_sequences
```

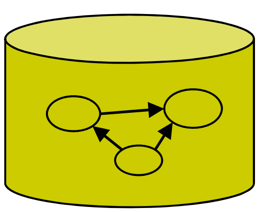
logical data independence



relations

```
SELECT seq
  FROM ncbi_sequences
 WHERE seq = 'GATTACGATATTA';
```

physical data independence



files and pointers

```
f = fopen('table_file');
fseek(10030440);
while (True) {
  fread(&buf, 1, 8192, f);
  if (buf == GATTACGATATTA) {
    . . .
```

Key Idea: Indexes

- Databases are especially, but exclusively, effective at "Needle in Haystack" problems:
  - **Extracting small results from big datasets**
  - **Transparently provide "old style" scalability**
  - **Your query will always* finish, regardless of dataset size.**

  - **Indexes are <u>easily built</u> and <u>automatically used</u> when appropriate**

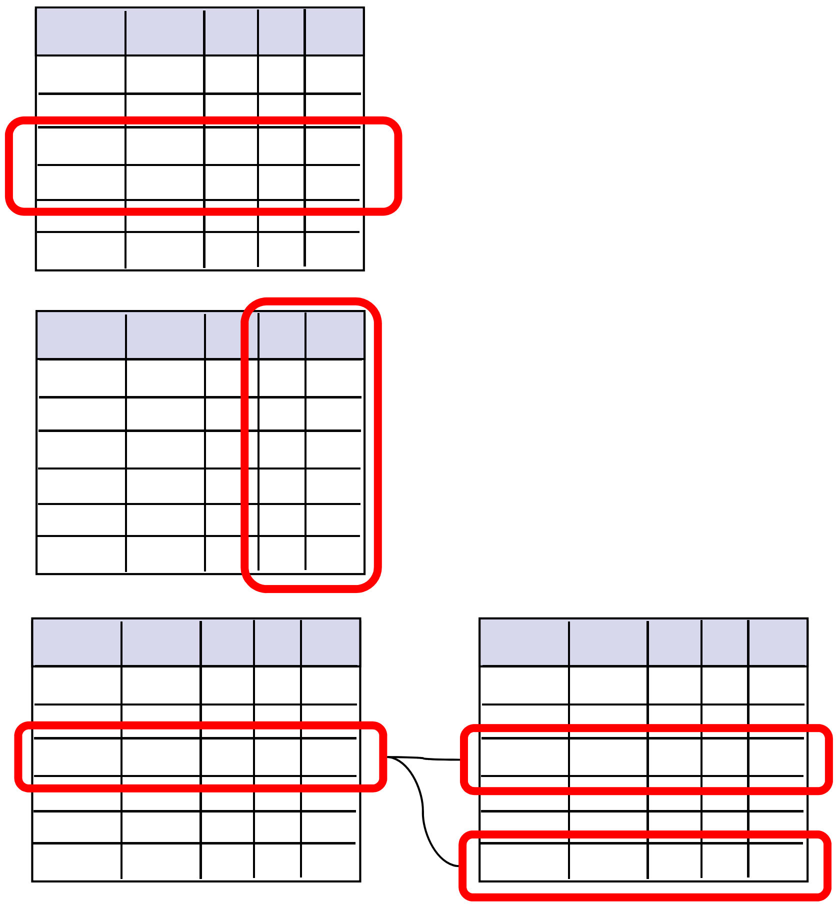
```
CREATE INDEX seq_idx ON sequence(seq);

SELECT seq
  FROM sequence

 WHERE seq = 'GATTACGATATTA';
```

***almost**

Key Idea: Bring the Computation to the Data

- It is no longer feasible to have users download data to their desktop for processing
- QLs push the computation down into the database
- Natural extension: data born in the cloud ands stays there forever
  - Computation comes and goes

Key Idea: An ***Algebra of Tables***



select

project

join

Other operators: aggregate, union, difference, cross product

Key Idea: Algebraic Optimization

N = ((z*2)+((z*3)+0))/1

Algebraic Laws:

1. (+) identity: x+0 = x
2. (/) identity: x/1 = x
3. (*) distributes: (n*x+n*y) = n*(x+y)
4. (*) commutes: x*y = y*x

Apply rules 1, 3, 4, 2:

N = (2+3)*z

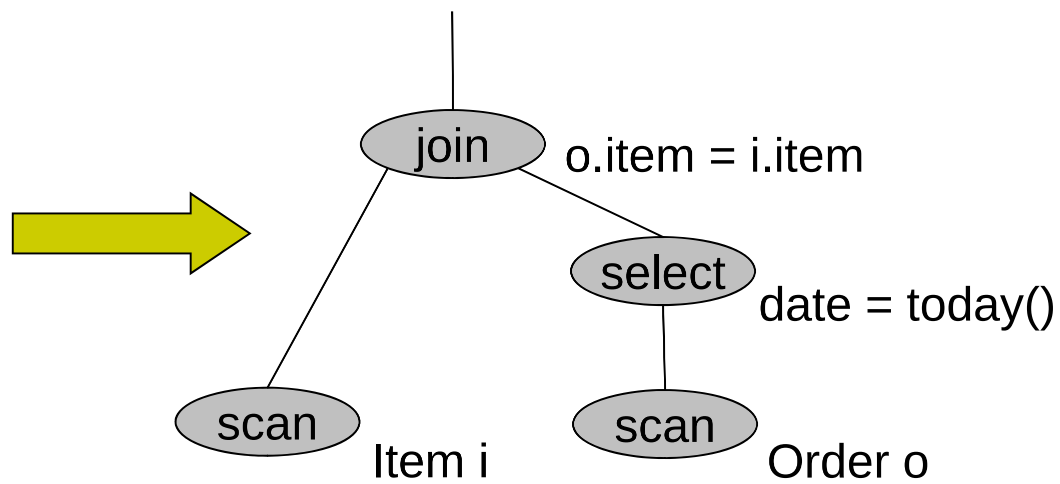
two operations instead of five, no division operator

Same idea works with the Relational Algebra!

Key Idea: Declarative Languages

Find all orders from today, along with the items ordered

```
SELECT *
  FROM Order o, Item i
 WHERE o.item = i.item
   AND o.date = today()
```



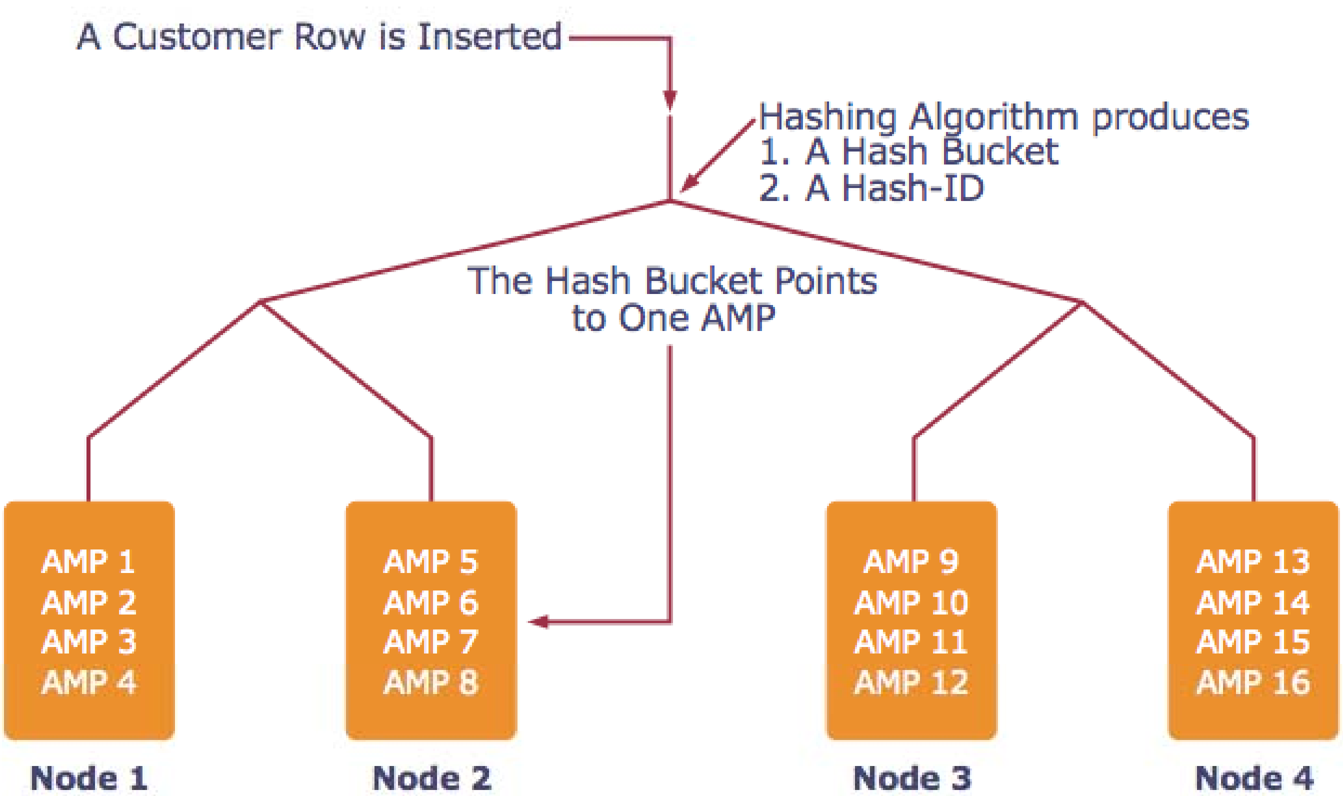
Key Idea: User-defined types and functions

```
SELECT salinity(temperature, conductivity) FROM CTD

SELECT * FROM FetchCTD(today())

SELECT date, median(salt) FROM CTD GROUP BY date
```

Current Topics

- Parallel
  - "SQL is the world's most popular parallel language"
  - Teradata, Greenplum, Netezza, Aster, Datallegro (now owned by MS)
- Column-oriented
  - Vertica, MonetDB
- Array-oriented
  - SciDB, RasDaMan
- Trees, Meshes, Graphs, Streams, ….
- Schema-free or schema-later approaches

Example Parallel System: Teradata

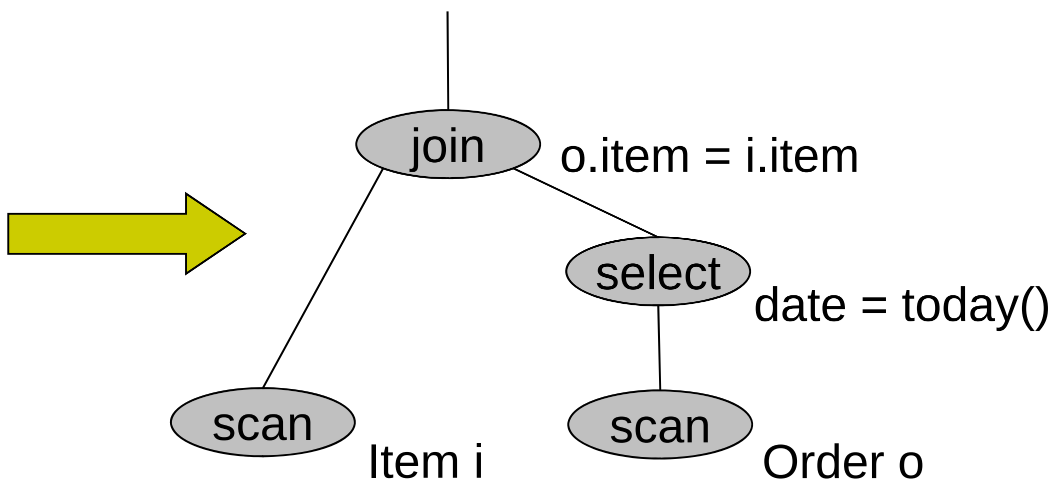


AMP = unit of parallelism

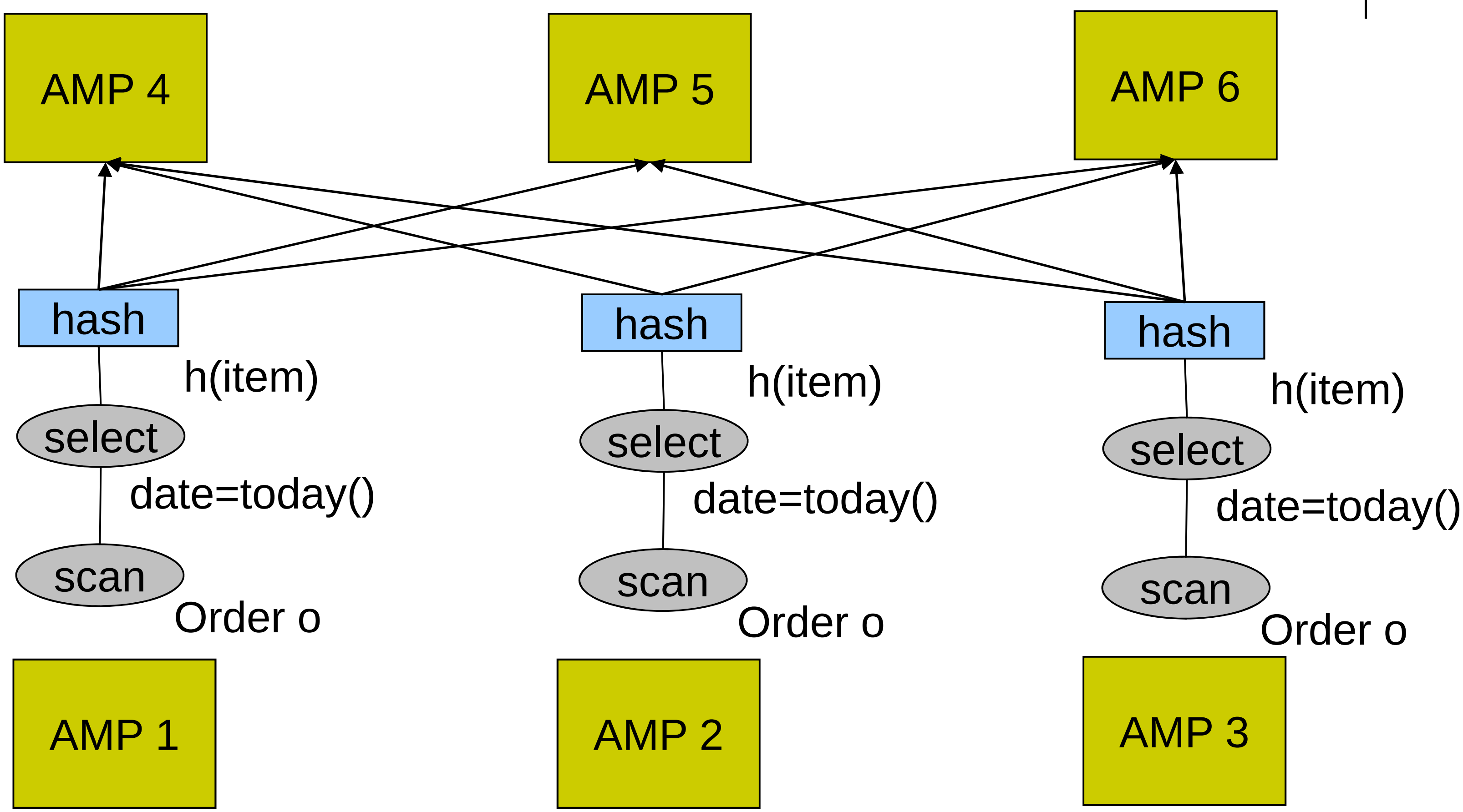
Example System: Teradata

Find all orders from today, along with the items ordered

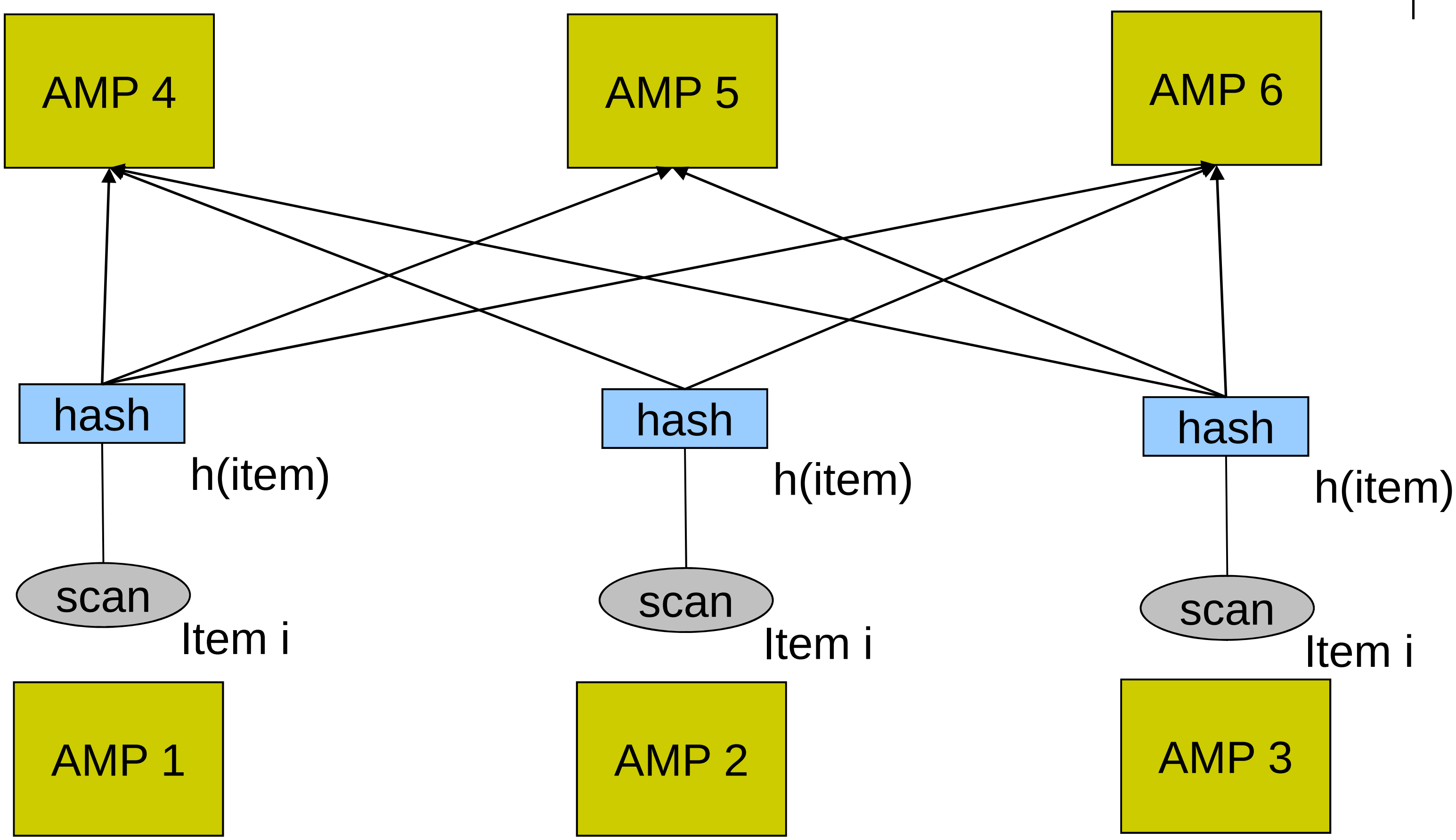
```
SELECT *
  FROM Orders o, Lines i
 WHERE o.item = i.item
   AND o.date = today()
```



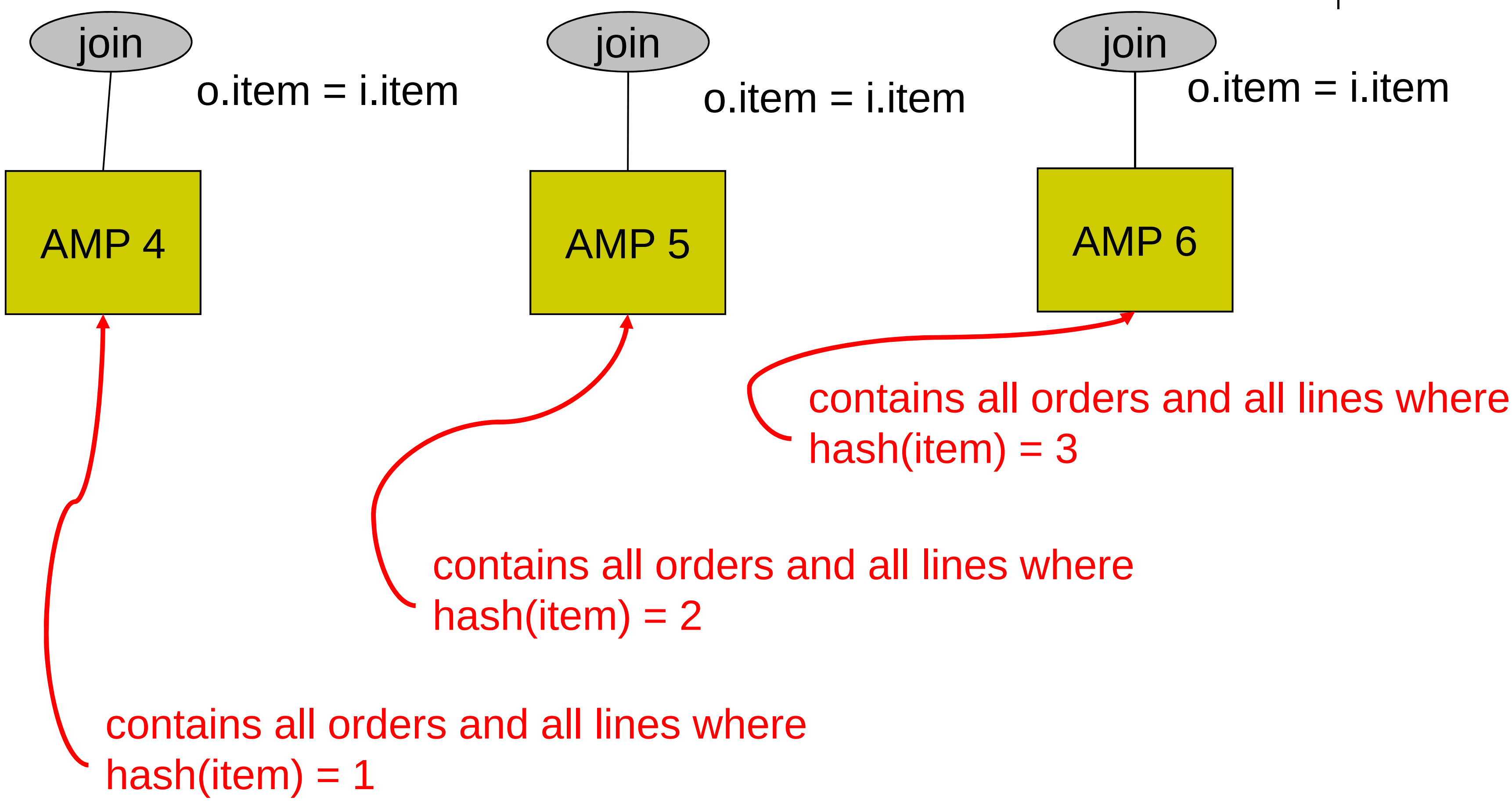
Example System: Teradata



Example System: Teradata



Example System: Teradata



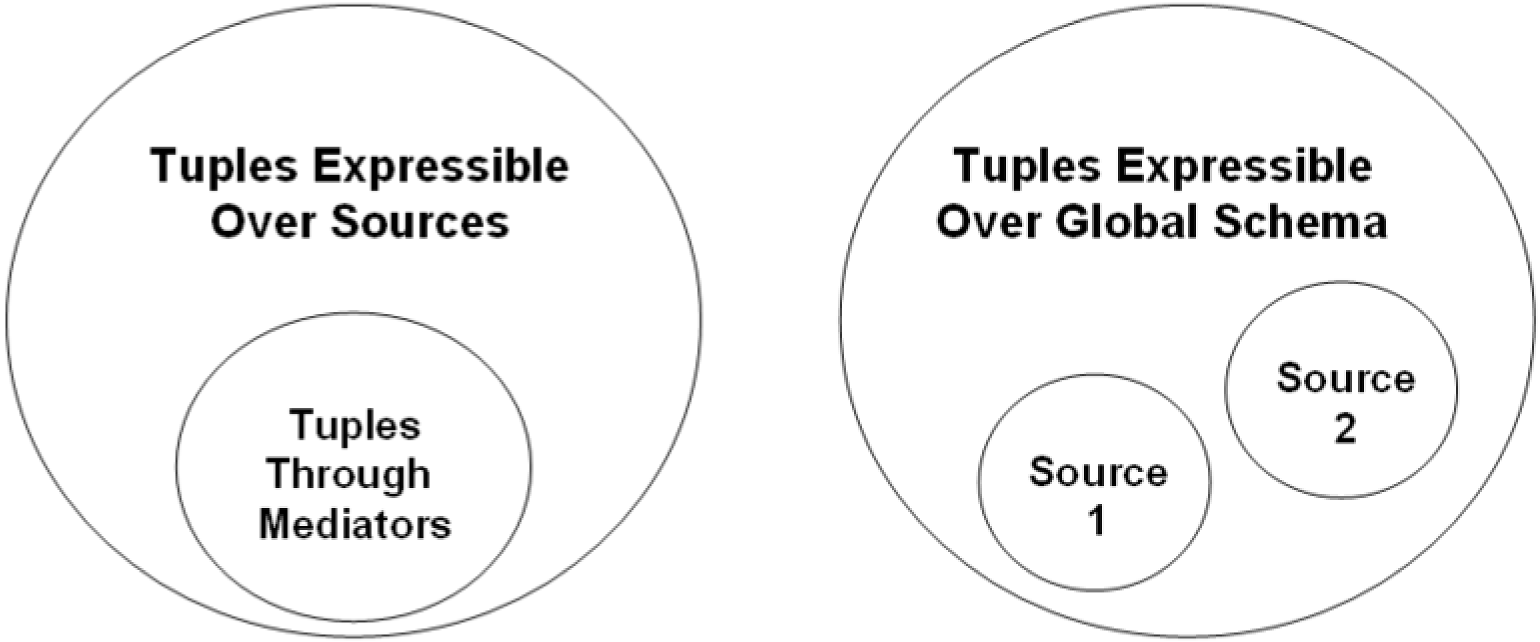
Views

- Push logic into the database
  - Single point of control
  - Encourages reuse
  - One problem: no source control
- Deployment Tool
  - Implement legacy schema with views
  - Legacy applications can be rewired incrementally

Views applied

- Views for legacy schemas
  - bontemp = temperature data at Bonneville Dam
- Views for simplifying php code
  - "forecast dashboard"
    - Complex aggregates, filters, transformation
    - Hide the complex SQL behind a view
- Views for the integrated schema
  - GAV vs. LAV both conceptually valid
  - Reality: GAV leads to performance problems
  - Union is an expensive operator; Postgresql has no parallel query evaluation.

GAV vs. LAV



Global As View

Local As View

Materialized Views

- What's a view?
  - A named query
  - re-executed at runtime
  - may be compiled

cheaters!

- Oracle and IBM obliterated TPC-D benchmarks in late 1998
- Oracle offered $1M to anyone who could show that MS SQL Server was within two orders of magnitude of Oracle on TPC-D

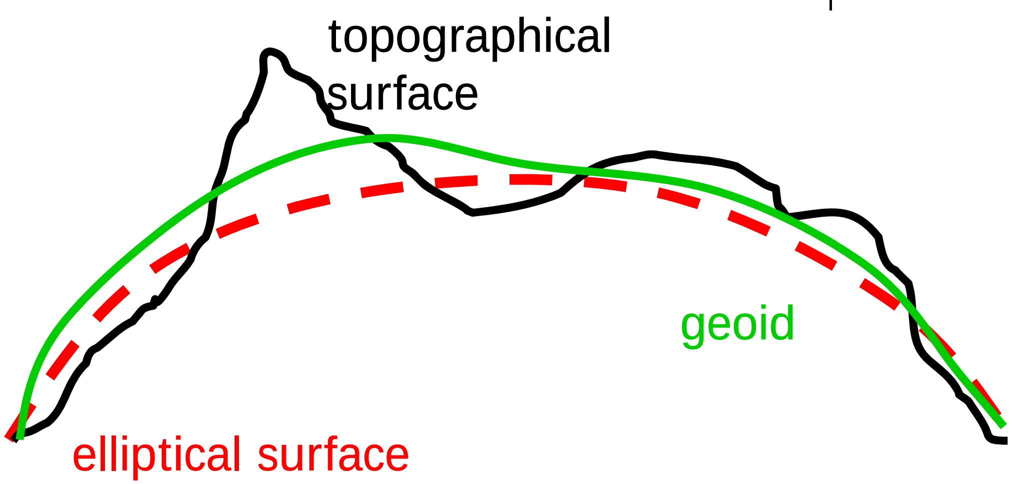
Materialized Views (cont'd)

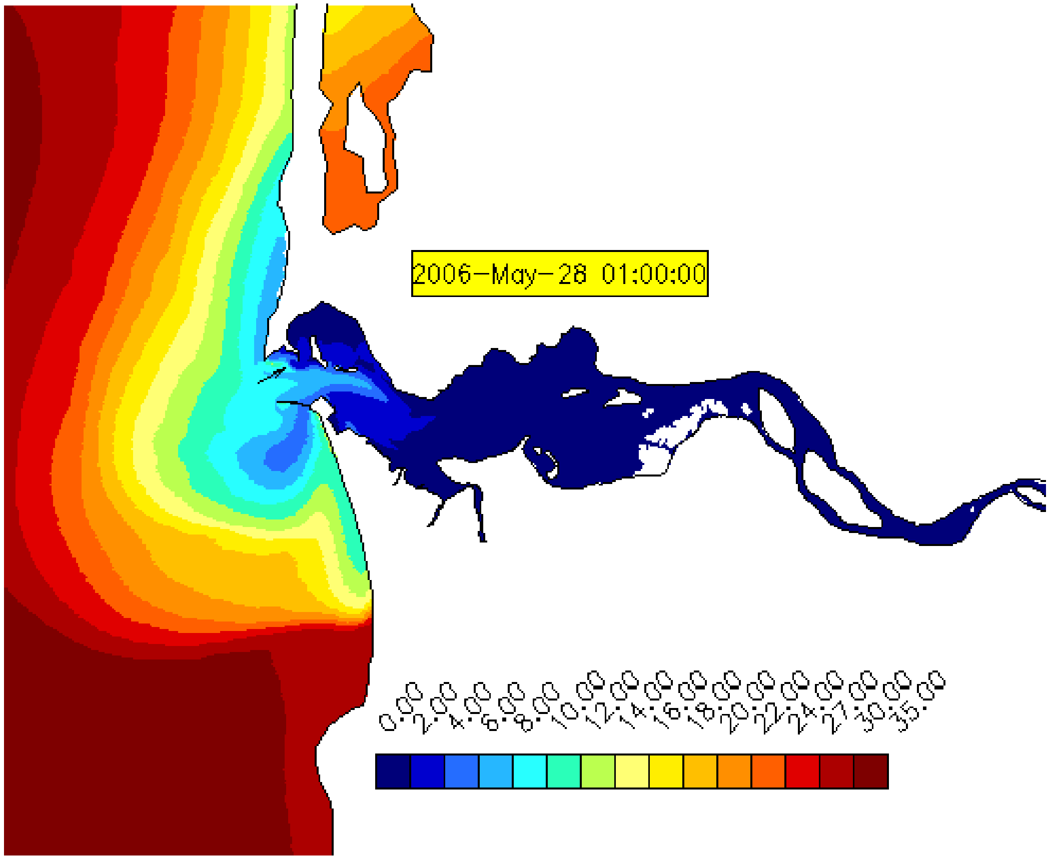
- How are they implemented in PostgreSQL?
  - Ordinary tables,
  - populated by rules, triggers, maybe cron jobs
- At CMOP:
  - Integrated Schema is a "Materialized View" over the legacy schemas
  - Populated by rules on the legacy schemas

Question: What's the difference between a Rule and a Trigger?

Geography (e.g., PostGIS)

- Three Contributions:
  - User Defined Types
    - Point
    - Line
    - Polygon
    - more
  - User Defined Functions
    - x(g), y(g), bbox(g), centroid(g), , distance(g,g), etc.
    - overlaps(g,g), contains(g,g), etc.
  - Geo-referencing





2006-May-28 01:00:00
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
20.00
22.00
24.00
27.00
30.00
35.00